

WE CLAIM:

1. Apparatus for processing data comprising:
 - 5 processing logic operable to perform data processing operations; and
 - an instruction decoder operable to decode program instructions to control said processing logic to perform data processing operations specified by said program instructions;
 - wherein said instruction decoder is responsive to a compare and branch
 - 10 instruction:
 - (i) to perform a comparison between a first value stored in a first register and a second value stored in a second register;
 - (ii) to determine a target branch address from a pre-programmed stored value; and
 - (iii) to branch to a sub-routine at said target branch address in dependence upon a
 - 15 result of said comparison.
2. Apparatus as claimed in claim 1, wherein said instruction is an array bounds checking instruction and said sub-routine is an array bounds exception handling routine.
- 20 3. Apparatus as claimed in claim 1, wherein at least one of said first register and said second register are specified within said compare and branch instruction.
4. Apparatus as claimed in claim 2, wherein said first value is a reference value
- 25 specifying an array size and said second value is a test value determined from a decoded program instruction.
5. Apparatus as claimed in claim 4, wherein said comparison determines whether said reference value is greater than or equal to said test value.
- 30 6. Apparatus as claimed in claim 4, wherein said result of said comparison is determined from a carry flag value and zero flag value.

7. Apparatus as claimed in claim 2, wherein said branching operation comprises copying a pointer to said array bounds exception handling routine into a register specifying a next program instruction.
- 5 8. Apparatus as claimed in claim 1, wherein said data processing apparatus comprises a co-processor and said pre-programmed stored value is read from a register of said co-processor.
9. Apparatus as claimed in claim 1, wherein said compare and branch instruction
10 is executed within a single processing cycle of said data processing apparatus when the branch is not taken.
10. Apparatus as claimed in claim 1, wherein said instruction decoder is operable to decode translated platform-independent program instructions.
- 15 11. Apparatus as claimed in claim 10, wherein said platform independent program instructions are one of :
Java bytecodes;
.net bytecodes;
20 MSIL bytecodes; and
CIL bytecodes.
12. Apparatus as claimed in claim 1, wherein said data processing apparatus is operable in a user mode and a privileged mode and said data processing apparatus
25 remains in said user mode during execution of said compare and branch instruction.
13. A method of processing data with an apparatus for processing data having processing logic operable to perform data processing operations and an instruction decoder operable to decode program instructions to control said processing logic to
30 perform data processing operations specified by said program instructions, said method comprising the steps of:
in response to a compare and branch instruction decoded by said instruction decoder controlling said processing logic:

- (i) to perform a comparison between a first value stored in a first register and a second value stored in a second register;
- (ii) to determine a target branch address from a pre-programmed stored value; and
- (iii) to branch to a sub-routine at said target branch address in dependence upon a
5 result of said comparison.

14. A method as claimed in claim 13, wherein said instruction is an array bounds checking instruction and said sub-routine is an array bounds exception handling routine.

10

15. A method as claimed in claim 13, wherein at least one of said first register and said second register are specified within said compare and branch instruction.

16. A method as claimed in claim 14, wherein said first value is a reference value
15 specifying an array size and said second value is a test value determined from a decoded program instruction.

17. A method as claimed in claim 16, wherein said comparison determines whether said reference value is greater than or equal to said test value.

20

18. A method as claimed in claim 16, wherein said result of said comparison is determined from a carry flag value and a zero flag value.

19. A method as claimed in claim 14, wherein said branching operation comprises
25 copying a pointer to said array bounds exception handling routine into a register specifying a next program instruction.

20. A method as claimed in claim 13, wherein said data processing apparatus comprises a co-processor and said pre-programmed stored value is read from a
30 register of said co-processor.

21. A method as claimed in claim 13, wherein said compare and branch instruction is executed within a single processing cycle of said data processing apparatus when the branch is not taken.

22. A method as claimed in claim 13, wherein said instruction decoder is operable to decode translated platform-independent program instructions.

5 23. A method as claimed in claim 22, wherein said platform independent program instructions are one of :

Java bytecodes;

.net bytecodes;

MSIL bytecodes; and

10 CIL bytecodes.

24. A method as claimed in claim 13, wherein said data processing apparatus is operable in a user mode and a privileged mode and said data processing apparatus remains in said user mode during execution of said compare and branch instruction.

15

25. A computer program product including a computer program operable to control an apparatus for processing data having processing logic operable to perform data processing operations and an instruction decoder operable to decode program instructions to control said processing logic to perform data processing operations
20 specified by said program instructions, said computer program comprising:

a compare and branch instruction decodable by said instruction decoder to control said processing logic:

(i) to perform a comparison between a first value stored in a first register and a second value stored in a second register;

25 (ii) to determine a target branch address from a pre-programmed stored value; and

(iii) to branch to a sub-routine at said target branch address in dependence upon a result of said comparison.

26. A computer program product as claimed in claim 25, wherein said instruction
30 is an array bounds checking instruction and said sub-routine is an array bounds exception handling routine.

27. A computer program product as claimed in claim 25, wherein at least one of said first register and said second register are specified within said compare and branch instruction.
- 5 28. A computer program product as claimed in claim 26, wherein said first value is a reference value specifying an array size and said second value is a test value determined from a decoded program instruction.
29. A computer program product as claimed in claim 28, wherein said comparison
10 determines whether said reference value is greater than or equal to said test value.
30. A computer program product as claimed in claim 28, wherein said result of said comparison is determined from a carry flag value and a zero flag value.
- 15 31. A computer program product as claimed in claim 26, wherein said branching operation comprises copying a pointer to said array bounds exception handling routine into a register specifying a next program instruction.
32. A computer program product as claimed in claim 25, wherein said data
20 processing apparatus comprises a co-processor and said pre-programmed stored value is read from a register of said co-processor.
33. A computer program product as claimed in claim 25, wherein said compare
25 and branch instruction is executed within a single processing cycle of said data processing apparatus when the branch is not taken.
34. A computer program product as claimed in claim 25, wherein said instruction decoder is operable to decode translated platform-independent program instructions.
- 30 35. A computer program product as claimed in claim 34, wherein said platform independent program instructions are one of :
Java bytecodes;
.net bytecodes;
MSIL bytecodes; and

CIL bytecodes.

36. A computer program product as claimed in claim 25, wherein said data processing apparatus is operable in a user mode and a privileged mode and said data processing apparatus remains in said user mode during execution of said compare and branch instruction.

37. A computer program product including a computer program operable to translate non-native program instructions to form native program instructions directly decodable by an apparatus for processing data having processing logic operable to perform data processing operations and an instruction decoder operable to decode program instructions to control said processing logic to perform data processing operations specified by said program instructions, said native program instructions comprising:

15 a compare and branch instruction decodable by said instruction decoder to control said processing logic:

- (i) to perform a comparison between a first value stored in a first register and a second value stored in a second register;
- (ii) to determine a target branch address from a pre-programmed stored value; and
- 20 (iii) to branch to a sub-routine at said target branch address in dependence upon a result of said comparison.

38. A computer program product as claimed in claim 37, wherein said instruction is an array bounds checking instruction and said sub-routine is an array bounds exception handling routine.

39. A computer program product as claimed in claim 37, wherein at least one of said first register and said second register are specified within said compare and branch instruction.

40. A computer program product as claimed in claim 38, wherein said first value is a reference value specifying an array size and said second value is a test value determined from a decoded program instruction.

41. A computer program product as claimed in claim 40, wherein said comparison determines whether said reference value is greater than or equal to said test value.
42. A computer program product as claimed in claim 40, wherein said result of
5 said comparison is determined from a carry flag value and a zero flag value.
43. A computer program product as claimed in claim 38, wherein said branching operation comprises copying a pointer to said array bounds exception handling routine into a register specifying a next program instruction.
10
44. A computer program product as claimed in claim 37, wherein said data processing apparatus comprises a co-processor and said pre-programmed stored value is read from a register of said co-processor.
- 15 45. A computer program product as claimed in claim 37, wherein said compare and branch instruction is executed within a single processing cycle of said data processing apparatus when the branch is not taken.
46. A computer program product as claimed in claim 37, wherein said instruction
20 decoder is operable to decode translated platform-independent program instructions.
47. A computer program product as claimed in claim 46, wherein said platform independent program instructions are one of :
Java bytecodes;
25 .net bytecodes;
MSIL bytecodes; and
CIL bytecodes.
48. A computer program product as claimed in claim 37, wherein said data
30 processing apparatus is operable in a user mode and a privileged mode and said data processing apparatus remains in said user mode during execution of said compare and branch instruction.